An Algorithm To Measure The Tightness Of A Bell Inequality

Author: Neil Smith

Supervisors: Dr Mark Howard and Dr Pieter Kok

Dec 2017

Abstract

Extending on previous work, we further develop an algorithm that measures the degree to which a Bell inequality is tight. We generalise the algorithm to work for any Bell inequality of any scenario where the parties can choose from different numbers of measurements and observe different numbers of outcomes for each of these measurements. We explain the theory behind the algorithm in depth and demonstrate the accuracy of the expressions by applying them to the CHSH inequality.

Contents

1 Prelude

In this paper I extended upon previous work [6] on the topic of non-locality, where I created an algorithm which calculates a measure of the tightness of a given Bell inequality by calculating the maximum classical bound and then the degree of intersection of the corresponding plane with the extremal points of the local polytope. The more extremal behaviours this plane intersects with, the more it aligns with a facet of the polytope. We call this measure of tightness the facet dimension of the Bell inequality. Tight Bell inequalities will intersect with the most extremal points and therefore have the highest facet dimension; the facet dimension will equal the spatial dimension of the Bell inequality.

In the field of non-locality we study how particles can behave in ways that cannot be explained by a local hidden variable (LHV) theory. Bell inequalities are expressions that characterise the correlations that can be observed between the outcomes of measurements on particles assuming nature behaved according to a LHV theory. In reality, it is possible to observe correlations beyond those possible in a LHV theory and therefore violate these Bell inequalities. We call these correlations, non-local correlations.

This non-local behaviour arises in entangled systems and can be exploited as a resource to perform tasks more efficiently in new quantum technologies. The more non-locality the system exhibits the more useful it is. Therefore, it is important to be able to quantify and determine how much non-locality is possible within a given entangled system. Indeed, one of the main open problems in the field of non-locality is determining the maximum possible correlation that could occur within a given system. My work deals with the problem of quantifying the degree of non-locality. In order to do this, one first needs to know the maximum classical correlation possible. This itself is difficult to determine and has only been done for the simplest Bell inequalities. In order to calculate the maximum classical correlation, the facets of the polytope of the local set of correlations must be determined.

In my previous work I created an algorithm that takes a given Bell inequality and calculates the maximum possible classical correlation and the extremal behaviours that result in this correlation. The number of these extremal behaviours that are linearly independent is a measure of how much the plane of the Bell inequality aligns with the facet of the polytope. Here, we call this quantity the facet dimension of the Bell inequality.

However, there were some drawbacks to the algorithm which I have now addressed in this work, these are:

1. The algorithm only worked for Bell inequalities that could be expressed in terms of correlators.

2. The algorithm only worked for scenarios in which each party has an equal number of measurements to choose from and there are an equal number of possible outcomes for each of these measurements.

The goal of this work was to modify the algorithm so that it can deal with an arbitrary Bell inequality as follows:

1. The algorithm should work for Bell inequalities that can be expressed in terms of probabilities.

2. The algorithm should work for scenarios in which all the parties have different numbers of measurements to choose from and have different numbers of outcomes they could observe for each of these measurements.

As far as we know, this is the first algorithm to generalise the calculation of the classical bound and calculate this quantity.

In section 2 we remind the reader of the relevant concepts and terminology used and explain how the theory generalises. In section 3 we discuss the changes in the algorithm and how the algorithm deals with such general scenarios. We explain how the input of the algorithm has now changed and its form and how we now calculate the relevant quantities. We provide a more in-depth explanation of how the algorithm calculates the maximum classical correlation and the quantities needed for the calculation of a dimension. We include a flowchart of the main method of the algorithm. In section 4 we explain which inequalities the algorithm was tested against reveal the results of the testing. In section 5 we conclude our work and discuss possible further work on the algorithm and how we now intend to apply the algorithm. In the appendix we demonstrate how these general expressions simplfy to their previous form in order to convince the reader of their correctness. Note that some of the notation has changed since our previous work and the reader should ignore previous definitions and follow only those provided in this work.

2 Preliminaries

2.1 Nonlocality

Here we explain how the framework in which we study non-locality generalises. There are n parties who each have access to their own system upon which they can make measurements and observe outcomes. But now, each party k can make a measurement m_k from a set of possible measurements $\mathcal{M}_k = \{1, 2, ..., M_k\}$ out of a total possible M_k measurements. When party k makes measurement m_k they can observe one outcome $d_{m_k,k}$ from a set of possible outcomes $\mathcal{D}_{m_k,k} = \{1, 2, ..., D_{m_k,k}\}$ out of a total possible $D_{m_k,k}$ outcomes. Any party k can also choose not to perform a measurement. We can think of this as them choosing to perform the zeroth measurement, which we denote as $m_k = 0$. The particular form of the measurements the parties perform and outcomes they observe are unimportant, only the number of measurements and outcomes are important.

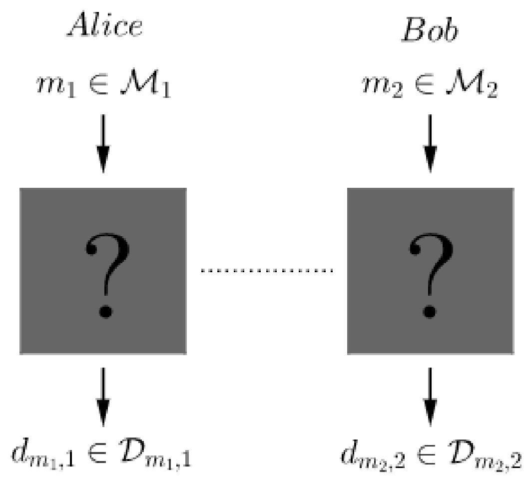


Figure 1: The Alice and Bob scenario, there are two parties who are very far apart from each other and each have access to a system upon which they can make measurements and observe outcomes. Alice can make a measurement m_1 from a set of $\mathcal{M}_1$ possible measurements and observe an outcome $d_{m_1,1}$ from a set of $\mathcal{D}_{m_1,1}$ possible outcomes. Similarly, Bob can make a measurement m_2 from a set $\mathcal{M}_2$ of possible measurements and observe an outcome $d_{m_2,2}$ from a set $\mathcal{D}_{m_2,2}$ of possible outcomes.

The joint probability that party k makes a measurement m_k on their system and observes an outcome $d_{m_k k}$ is now denoted $P(d_{m_1 1} d_{m_2 2} ... d_{m_n n} | m_1 m_2 ... m_n)$. Like before, we can also define a joint probability distribution that can be represented as a vector (1), which is known as a behaviour. The number of elements of this vector will just be the total number of possible ways of getting different outcomes. This can be found by calculating for each party the sum of the total number of possible outcomes for each of their measurements and then taking the product of all these numbers.

$$\vec{P}(d_{m_1,1} d_{m_2,2} ... d_{m_n,n} | m_1 m_2 ... m_n) \in \mathbb{R}^b \qquad b = \prod_{k=1}^{n} \left(\sum_{m_k=1}^{M_k} D_{m_k,k} \right) \tag{1}$$

2.2 The Local Polytope

The extremal behaviours (2) are those behaviours or probability distributions that can be expressed as a deterministic product of the local probability distributions of each party.

$$P(d_{m_1,1} d_{m_2,2} ... d_{m_n,n} | m_1 m_2 ... m_n) = D(d_{m_1,1}|m_1) D(d_{m_2,2}|m_2) ... D(d_{m_n,n}|m_n) \tag{2}$$

As shown in my previous work it is possible to determine the exact form of the extremal behaviours by combining the deterministic property of the behaviours with the no-signalling and normalisation conditions into a single set of constraints. Now that the parties can choose from different numbers of measurements and observe different numbers of outcomes this set of constraints is now modified to (3).

$$\sum_{d_{m_k,k}} D_k(d_{m_k,k}|m_k) = 1 \quad \forall\, m_k \in \mathcal{M}_k \quad \forall\, k \in \{1, ..., n\} \tag{3}$$

The total number of these constraints is denoted n_{con} and is given by (4).

$$n_{con} = \sum_{k=1}^{n} M_k \tag{4}$$

Using the same argument as I demonstrated in my previous work, we can determine from the constraints (3) the total number of extremal behaviours. We denote this number by n_{ex} and this is given by (5).

$$n_{ex} = \prod_{k,m_k} D_{m_k,k} \tag{5}$$

The dimension of the local polytope is now calculated according to (6) as shown in Theorem 3.1 of [5].

$$\dim \mathcal{L} = \prod_{k=1}^{n} \left(\sum_{m_k=1}^{M_k} (D_{m_k,k} - 1) + 1 \right) - 1 \tag{6}$$

2.3 Bell Inequalities

Previously, we only considered Bell inequalities that could be expressed as a sum of correlators. Now, we consider general Bell inequalities that can be expressed as a sum of probabilities $P_{i_j,j}$ with coefficients $c_{i_j,j}$ as shown in (7). Here, each term is denoted with two indices for reasons that will become clear later.

$$S = \sum_{i,j} c_{i_j,j} P_{i_j,j} = \vec{P} \cdot \vec{c} \leq S_{max} \tag{7}$$

Like before, these Bell inequalities can also be expressed geometrically as a series of planes in a space of probabilities and express these Bell inequalities as an inner product of a vector of probabilities $\vec{P}$ that defines the space with a vector of coefficients $\vec{c}$. The vector of coefficients determines only the orientation of the corresponding set of planes of the Bell Inequality.

Again, like before, terms can arise in Bell Inequalities where one or more of the parties do not make any measurements on their system. In this case, the joint probabilities P_i will be replaced with the marginal over the parties that do not make measurements. If there are $n' < n$ parties that make measurements then the joint probability will be a deterministic product of these n' parties that do make measurements. Suppose only party k does not make a measurement, then the marginal probability[1] would be expressed as (8).

$$\begin{aligned} &P(d_{m_1 1}...d_{m_{k-1},k-1} d_{m_{k+1},k+1}...d_{m_n,n}|m_1...m_{k-1}m_{k+1}...m_n) = \\ &\qquad D(d_{m_1,1}|m_1)...D(d_{m_{k-1},k-1}|m_{k-1}) D(d_{m_{k+1},k+1}|m_{k+1})...D(d_{m_n,n}|m_n) \end{aligned} \tag{8}$$

As mentioned earlier, each party k can also choose not to perform a measurement $m_k = 0$. Therefore, the total number of possible combinations of measurements the parties could choose to perform, or measurement

[1] Because terms like these marginal probabilities can arise, in general, there may not be a unique way in which to express a particular Bell Inequality.

settings, is equal to that in (9), and is denoted n_{set}.

$$n_{set} = \prod_{k=1}^{n} (M_k + 1) \tag{9}$$

The dimension of the facet Bell Inequalities (10) will, also like before, be one less than the dimension of the polytope (6).

$$\dim \mathcal{I} = \prod_{k=1}^{n} \left(\sum_{m_k=1}^{M_k} (D_{m_k,k} - 1) + 1 \right) - 2 \tag{10}$$

At this point I would like to address an issue in my previous work. At the time it was thought that, in general, the Bell Inequalities could have a spatial dimension less than that of the facet dimension. It is now thought that this is not true and every non-trivial Bell inequality does in fact have the same spatial dimension, equal to that of the spatial dimension of the facet.

3 The Algorithm

The algorithm loops over all the possible extremal behaviours by varying the local probabilities and calculates the corresponding Bell value S of the left hand side of (7). It keeps track of the maximum Bell value it has found so far and the corresponding sets of local probabilities that give this maximum. The final maximum Bell value S_{max} will be the maximum classical bound of this Bell inequality. The algorithm then calculates the corresponding behaviours that give this maximum classical bound from the sets of local probabilities. The facet dimension is the number of linearly independent behaviours that lie on the corresponding plane of the inequality, which can be found by calculating the rank of the matrix of these behaviours.

Originally, the algorithm only worked for inequalities that could be expressed as a sum of correlators, and only for scenarios where all of the parties k can choose from an equal number of measurements $M_k = m$ and observe an equal number of possible outcomes $D_{m_k,k} = d$. The algorithm required four things as input in order to accomplish this:

1. The total number of parties involved n.
2. The number of measurements each party could make m.
3. The number of outcomes each party could observe for each of their measurements d.
4. A list of coefficients of correlators that specify the particular Bell inequality c_i.

Now, the algorithm requires only two things as input:

1. A list of the total number of outcomes each party could observe for their measurements $D_{m_k,k}$.
2. A list of coefficients $c_{i_j,j}$ of probabilities $p_{i_j,j}$ that specify the particular Bell inequality.

The list of number of outcomes $D_{m_k,k}$ will be 2-dimensional, the column specifies the party k and the row specifies the particular measurement m_k of that party. The list won't take the form of a simple matrix with dimensions Rows $\times$ Columns as the number of rows m_k of each column k depends on the column k. This list can be thought of as a list of column vectors of different sizes. In order to deal with this, a cell list structure is used. Information about both the number of parties involved n and the number of measurements each party can choose from M_k is contained within this list, they are the number of columns and rows respectively, and so no longer need to be specified independently.

$$D_{m_k,k} = \left(\begin{pmatrix} D_{1,1} \\ D_{2,1} \\ \vdots \\ D_{M_1,1} \end{pmatrix}, \begin{pmatrix} D_{1,2} \\ D_{2,2} \\ \vdots \\ D_{M_2,2} \end{pmatrix}, \cdots, \begin{pmatrix} D_{1,n} \\ D_{2,n} \\ \vdots \\ D_{M_n,n} \end{pmatrix} \right) \tag{11}$$

The list of coefficients of probabilities $c_{i_j,j}$ will also be 2-dimensional and have a form similar to that of the list of number of outcomes $D_{m_k,k}$. Each column j specifies the the measurement settings $\mathcal{M} = \{m_1, m_2, ..., m_n\}$ and the row i_j specifies a particular probability of getting one of the possible outcomes given the parties make those measurements. Since terms can arise where one or more of the parties do not make a measurement, the number of rows R_j also depends on the column j. For this reason, we also use a cell list structure for the list of probability coefficients. The corresponding list of probabilities $p_{i_j,j}$ will have the same dimension as $c_{i_j,j}$.

$$c_{i_j,j} = \left(\begin{pmatrix} c_{1,1} \\ c_{2,1} \\ \vdots \\ c_{R_1,1} \end{pmatrix}, \begin{pmatrix} c_{1,2} \\ c_{2,2} \\ \vdots \\ c_{R_2,2} \end{pmatrix}, \cdots, \begin{pmatrix} c_{1,J} \\ c_{2,J} \\ \vdots \\ c_{R_J,J} \end{pmatrix} \right) \tag{12}$$

The column j of a particular set of measurements $\mathcal{M} = \{m_1, m_2, ..., m_n\}$ can be found through (13). The total number of rows J can be found by subsituting $\mathcal{M} = \{M_1, M_2, ..., M_n\}$ into this expression.

$$j = \sum_{\substack{r=1 \\ m_r \in \mathcal{M}}}^{n} \left(m_r \prod_{k=r+1}^{n} (M_k + 1) \right) + 1 = m_n + m_{n-1}(M_n + 1) + m_{n-2}(M_{n-1} + 1)(M_n + 1) + ... + 1 \tag{13}$$

$j =$	1	2	...	$M_n + 1$	$M_n + 2$	...	J
$\mathcal{M} =$	$\{0, 0, ..., 0, 0\}$	$\{0, 0, ..., 0, 1\}$	...	$\{0, 0, ..., 0, M_n\}$	$\{0, 0, ..., 1, 0\}$	...	$\{M_1, M_2, ..., M_{n-1}, M_n\}$

Table 1: The number associated with the measurement settings $\mathcal{M}$ is being incremented as the row j increases. Each digit r of this number can have a different maximum value M_r.

The number of rows R_j of the jth column of $c_{i_j,j}$ can be found by taking the product of the total number of outcomes $D_{m_r,r}$ for each of these measurements m_r in $\mathcal{M}$ that are non-zero (A.8).

$$R_j = \prod_{\substack{r \\ \forall \{r \,|m_r \in \mathcal{M}\,, m_r \neq 0\}}} D_{m_r,r} \tag{14}$$

These concepts can be explained more easily through an example. Consider the CHSH inequality, in this scenario there are $n = 2$ parties involved and therefore $D_{m_k,k}$ has 2 columns. Each party can choose from only 2 measurements $M_1 = M_2 = m = 2$ and so the length of each column will be 2. Each party can only ever observe 2 possible outcomes when they make a measurement on their system so each element of the outcome list is 2, $D_{m_k,k} = 2 \,\forall\, m_k, k$.

$$D^{\text{CHSH}}_{m_k,k} = \left(\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right) \tag{15}$$

The total number of possible measurement settings found from (9) is 9 so there are 9 column vectors part of $c_{i_j,j}$ and $p_{i_j,j}$. The CHSH inequality expressed in terms of probabilities is given by (16).

$$\begin{gathered} P(11|11) - P(12|11) - P(21|11) + P(22|11) \\ P(11|12) - P(12|12) - P(21|12) + P(22|12) \\ P(11|21) - P(12|21) - P(21|21) + P(22|21) \\ -P(11|22) + P(12|22) + P(21|22) - P(22|22) \leq 2 \end{gathered} \tag{16}$$

The possible measurement settings in this scenario are $\mathcal{M} = \{0, 0\}$, $\{0, 1\}$, $\{0, 2\}$, $\{1, 0\}$, $\{1, 1\}$, $\{1, 2\}$, $\{2, 0\}$, $\{2, 1\}$, $\{2, 2\}$ and the corresponding form of the list of probabilities is shown in (17). The subscripts 1

and 2 are used to help clarify which party is making the measurement.

$$p_{i_j,j}^{\mathsf{CHSH}} = \left((1), \begin{pmatrix} P(1|1_2) \\ P(2|1_2) \end{pmatrix}, \begin{pmatrix} P(1|2_2) \\ P(2|2_2) \end{pmatrix}, \begin{pmatrix} P(1|1_1) \\ P(2|1_1) \end{pmatrix}, \begin{pmatrix} P(11|11) \\ P(12|11) \\ P(21|11) \\ P(22|11) \end{pmatrix}, \right.$$
$$\left. \begin{pmatrix} P(11|12) \\ P(12|12) \\ P(21|12) \\ P(22|12) \end{pmatrix}, \begin{pmatrix} P(1|2_1) \\ P(2|2_1) \end{pmatrix}, \begin{pmatrix} P(11|21) \\ P(12|21) \\ P(21|21) \\ P(22|21) \end{pmatrix}, \begin{pmatrix} P(11|22) \\ P(12|22) \\ P(21|22) \\ P(22|22) \end{pmatrix} \right) \tag{17}$$

From (16) we can see that terms only arise where all the parties make measurements and so the only non-zero columns of $c_{i_j,j}$ are those where both parties make a measurement $\mathcal{M} = \{1,1\},\{1,2\},\{2,1\},\{2,2\}$. The corresponding list of coefficients is given in (18).

$$c_{i_j,j}^{\mathsf{CHSH}} = \left((0), \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} \right) \tag{18}$$

As mentioned in my previous work, the algorithm loops over all the extremal behaviours by looping over the possible values of a set of numbers $\mathcal{I}$ which we call the index numbers, shown in (19).

$$\mathcal{I} = \{I_{1,1}, I_{1,2}, ..., I_{m_k,k}, ..., I_{M_n,n}\} \tag{19}$$

The extremal behaviours are those probability distributions that obey the constraints (3), the number of these constraints is given by (4). Each one of these constraints states that the local probabilities of getting the different outcomes for a particular measurement of a party must sum to unity. Since these local probabilities are deterministic probabilities, probabilities that can only be 0 or 1, only one of these probabilities can be 1 at any given time. Each constraint is independent of the others as no local probability occurs in more than one of the constraints. Together this tells us that the total number of extremal behaviours can be found by calculating the total number of ways of distributing the 1s amongst the probabilities in the constraints. We can even determine the exact form of these behaviours as we know the form of the local probabilities.

We can use a set of numbers $\mathcal{I}$ to denote which of the probabilities in each constraint is currently 1. We call these the index numbers, they index which of the probabilities are 1. There is one index number for each constraint, and like each constraint, each index number $I_{m_k,k}$ is denoted by the measurement m_k and the party k. The maximum number that these index variables can take is equal to the maximum number of outcomes for that particular measurement of that party $D_{m_k,k}$.

From the set of index numbers $\mathcal{I}$ we can calculate the local probabilities and therefore $p_{i_j,j}$ and the behaviours. All the extremal behaviours can be found by varying the values of these index numbers, the search space of the algorithm is represented as a tree diagram. Because the maximum number of these index variables can all be different, the branching factor in the search tree of the algorithm can be different at each depth as shown in Figure 2.

As an example, consider the index numbers for the CHSH inequality. There will be 4 index variables in total as there are 2 parties $n = 2$ and they can both make two measurements each $M_1 = M_2 = 2$, so $\mathcal{I}$ will take the form (20).

$$\mathcal{I}^{\mathsf{CHSH}} = \{I_{1,1}, I_{2,1}, I_{1,2}, I_{2,2}\} \tag{20}$$

There are two possible outcomes for each of these measurements $D_{m_k,k} = 2$ so there will be a total of 8 local probabilities which we can also denote as a set $\mathcal{P}_{local}$ as shown in (21). The dashed vertical lines are used

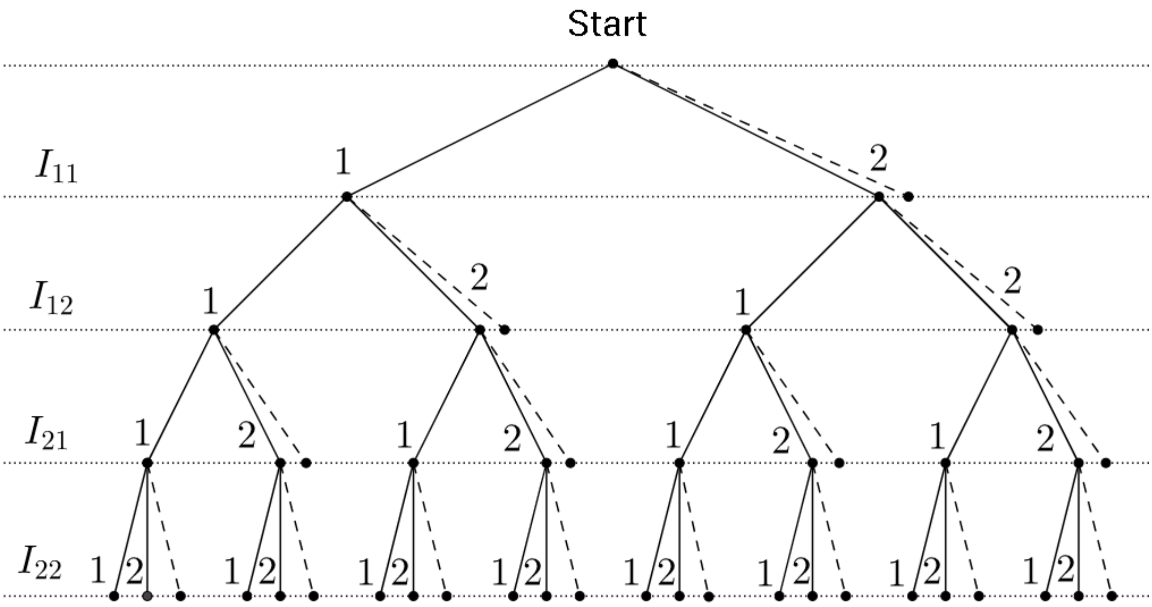


Figure 2: An example of the index number search space of the algorithm. The search space takes the form of a tree diagram. The branching factor at each depth can vary and is given by the numbers $D_{m_k,k}$. Each node represents a particular value of one of the index numbers. Nodes at the same depth represent different values of the same index number. The values of the index numbers at each final node can be used to determine the exact form of one of the extremal behaviours.

to separate the local probabilities that belong to different constraints.

$$\mathcal{P}_{local} = \{P(1|1_1), P(2|1_1) \,\vdots\, P(1|2_1), P(2|2_1) \,\vdots\, P(1|1_2), P(2|1_2) \,\vdots\, P(1|2_2), P(2|2_2)\} \; . \tag{21}$$

Table 2 lists some of the possible values of the CHSH index numbers $\mathcal{I}^{\mathsf{CHSH}}$ and the corresponding values of the local probabilities $\mathcal{P}_{local}$.

$\mathcal{I} =$	$\{1,1,1,1\}$	$\{1,1,1,2\}$	...	$\{2,2,2,2\}$
$\mathcal{P}_{local} =$	$\{10 \,\vdots\, 10 \,\vdots\, 10 \,\vdots\, 10\}$	$\{10 \,\vdots\, 10 \,\vdots\, 10 \,\vdots\, 01\}$	...	$\{01 \,\vdots\, 01 \,\vdots\, 01 \,\vdots\, 01\}$

Table 2: The index numbers $\mathcal{I}^{\mathsf{CHSH}}$ determine which local probabilities $\mathcal{P}_{local}$ are 1, the rest are 0.

Comments have also been added to the code and most of the variables and methods have been renamed for better readability. The new list of methods are given in Table 3. All the functions have just been generalised to deal with the general case. Instead of a method to calculate the correlator we now have a method to calculate the individual probabilities "calcProb".

Shown in Figure 3 is a flowchart of how the main method loopExtremalBehaviours loops over all the possible extremal behaviours by looping over all the possible values of the index numbers. For each value of the index numbers it loops over all the elements of the probability coefficient list *probCoeffList* and calculates the contribution to the Bell value. Once it has calculated the Bell value for this set of local probabilities *s*, it compares it to the maximum Bell value found so far *sMax* and then either stores this set of local probabilities or discards it. If it has found a new maximum then it discards all of the previous sets of local probabilities that gave the old maximum. At the end of this the program calculates the behaviours from the local probabilities and the facet dimension from these. A clone of the repository at github can be downloaded using https://github.com/s3nt1n3lz21/BellInequalityResearch.git

Method	Description
calc	Starts the calculation of the facet dimension and classical bound.
loopExtremalBehaviours	Loops over the extremal behaviours and calculates the Bell value.
getLocalProbIndex	Calculates the index of a particular local probability within the list of local probabilities.
calcProb	Calculates the probability the parties get a particular set of outcomes for the measurements they make.
calcProbDists	Calculates all the behaviours from the sets of local probabilities that give the classical bound.
calcProbDist	Calculates the behaviour of a given set of local probabilities.
calcDim	Calculates the dimension from the behaviours that give the classical bound.

Table 3: The methods part of the class which calculates the classical bound and facet dimension of a Bell Inequality.

4 Testing And Results

The algorithm was tested against eight tight inequalities. Tests 1-4 are the CHSH inequality [3], equation 19 in [4], equation 8 in [7] and example 4 in [1] respectively. Tests 5-7 are equations 11 and 12 in [2] and equation 10 in [4] respectively.

Tests 1-6 are all inequalities involve scenarios where the parties can all perform the same number of measurements and observe the same number of outcomes for each of these measurements. But now, Tests 5 and 6 test that the algorithm works for higher numbers of parties n. Test 7 is an inequality in a scenario where the parties can make different numbers of measurements.

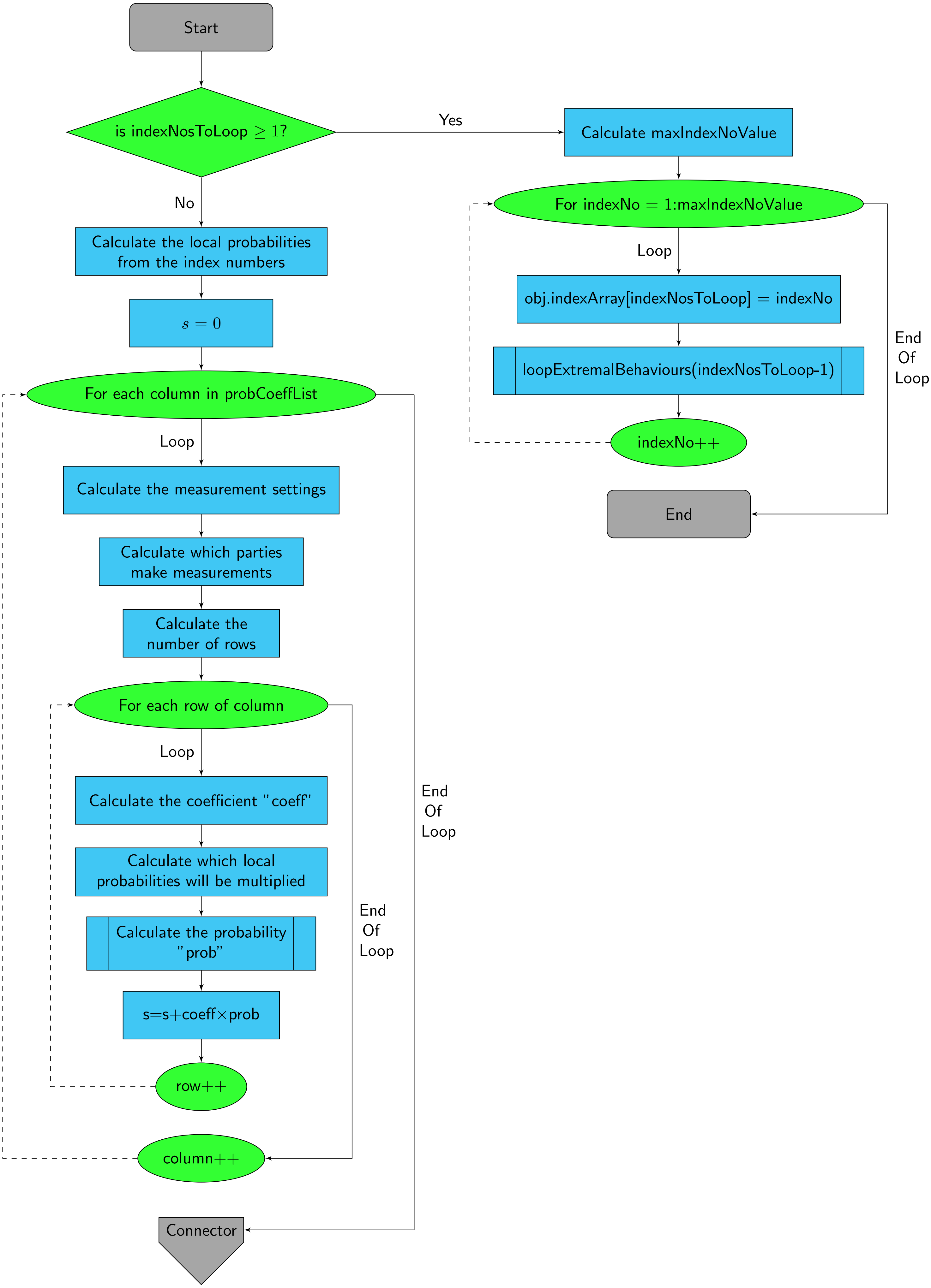
Test	$D_{m_k,k}$	Expected		Calculated		Result
		Dimension	Bound	Dimension	Bound	
1	((2;2),(2;2))	7	2	7	2	Pass
2	((2;2;2),(2;2;2))	14	0	14	0	Pass
3	((2;2),(2;2),(2;2))	25	6	25	6	Pass
4	((2;2;2),(2;2;2),(2;2;2))	62	8	62	8	Pass
5	((2;2),(2;2),(2;2),(2;2))	79	2	79	2	Pass
6	((2;2),(2;2),(2;2),(2;2),(2;2))	241	2	241	2	Pass
7	((2;2),(2;2;2))	10	0	10	0	Pass

Table 4: The Inequalities the program was tested against and the results.

The algorithm successfully produced the correct results for all of these inequalities where the input was a list of probability coefficients. This should convice the reader the algorithm works for scenarios where the parties can choose from different numbers of measurements. There is not much in the literature in terms of inequalities in scenarios in which the parties can observe different numbers of outcomes, this is because we are usually only interested in binary-outcome measurements. For this reason, the algorithm has not been tested against this case.

5 Conclusions And Further Work

Extending on previous work, we generalised our algorithm which calculates a measure of how tight a given Bell Inequality is. Previously our algorithm worked only for those inequalities that could be expressed in terms of correlators and only involved scenarios where the parties could choose from an equal number of measurements and observe the same number of outcomes for each of these measurements. Now, the algorithm has been


Start
is indexNosToLoop ≥ 1?
Yes
No
Calculate maxIndexNoValue
For indexNo = 1:maxIndexNoValue
Loop
obj.indexArray[indexNosToLoop] = indexNo
loopExtremalBehaviours(indexNosToLoop-1)
indexNo++
End Of Loop
End
Calculate the local probabilities from the index numbers
s = 0
For each column in probCoeffList
Loop
Calculate the measurement settings
Calculate which parties make measurements
Calculate the number of rows
For each row of column
Loop
Calculate the coefficient "coeff"
Calculate which local probabilities will be multiplied
Calculate the probability "prob"
s=s+coeff×prob
row++
End Of Loop
End Of Loop
column++
Connector

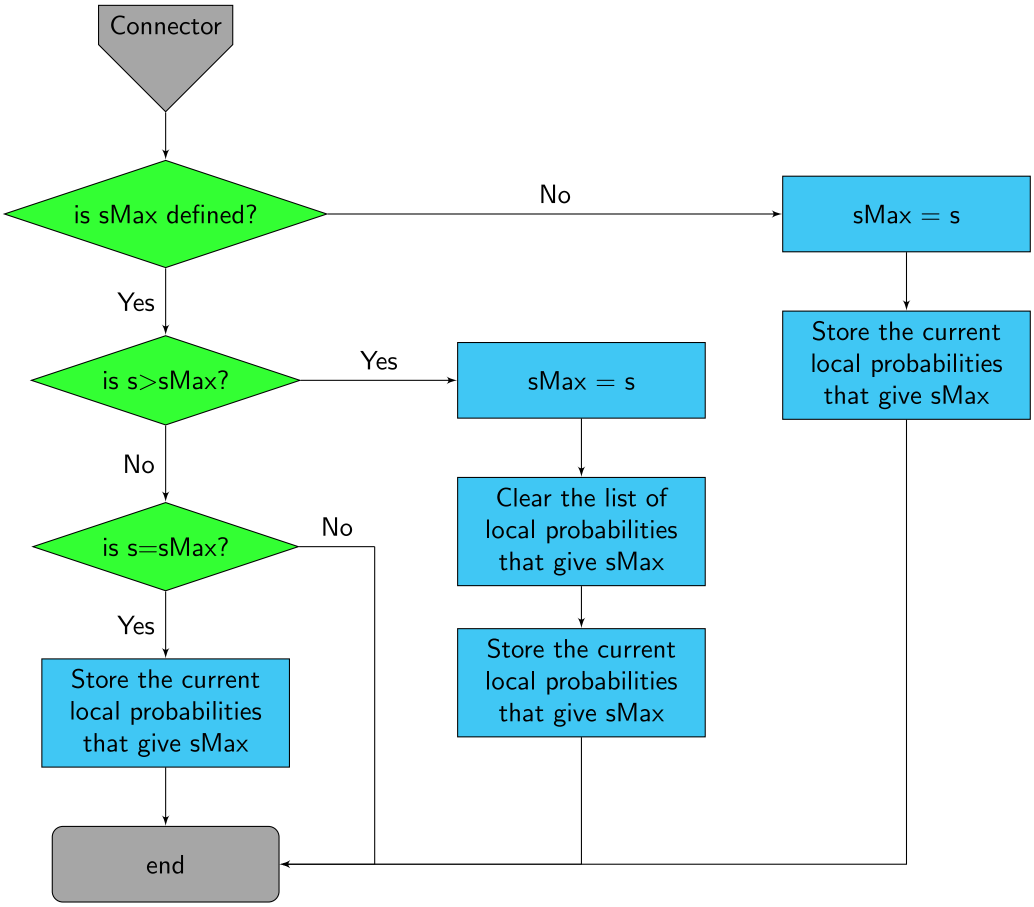


Figure 3: A flowchart of the function loopExtremalBehaviours which calculates the classical value achievable for each extremal behaviour.

developed to work for any Bell inequality in general scenarios where the parties can choose from different numbers of measurements and observe different numbers of outcomes for each of these measurements.

The algorithm was tested against an array of different tight Bell inequalities and successfully calculated the correct dimension and classical bound. It successfully calculates these quantities for inequalities in probability form and for inequalities where the parties can make different numbers of measurements. However, it still needs to be tested against inequalities involving scenarios with different numbers of outcomes.

In our next work, we are going to use another program which calculates a Bell inequality through robustness from a given quantum state and measurements and apply this algorithm to study how the facet dimension of these Bell inequalities varies for different states and measurements and search for useful tight Bell inequalities.

A Appendix

In order to convice the reader of the accuracy of these expressions, in this section, we will demonstrate that they reduce to their previous forms and produce the correct results for CHSH. We assume that all of the parties k can choose from $M_k = m$ measurements and observe $D_{m_k,k} = d$ possible outcomes for each of these measurements.

The dimensions of the behaviours are

$$b = \prod_{k=1}^{n} \left(\sum_{m_k=1}^{M_k} D_{m_k k} \right) = \prod_{k=1}^{n} \left(\sum_{m_k=1}^{m} d \right) = (md)^n \tag{A.1}$$

The number of constraints is

$$n_{con} = \sum_{k=1}^{n} M_k = \sum_{k=1}^{n} m = nm \tag{A.2}$$

The number of extremal behaviours is

$$n_{ex} = \prod_{k,m_k} D_{m_k,k} = \prod_{k=1}^{n}\left(\prod_{m_k=1}^{m} d\right) = d^{nm} \tag{A.3}$$

The dimension of the polytope is

$$\dim \mathcal{L} = \prod_{k=1}^{n}\left(\sum_{m_k=1}^{M_k}(D_{m_k k}-1)+1\right)-1 = \prod_{k=1}^{n}\left(\sum_{m_k=1}^{m}(d-1)+1\right)-1 = (m(d-1)+1)^n - 1 \tag{A.4}$$

Similarly, the dimension of the Bell inequalities is one less than this.

The number of measurement settings is

$$n_{set} = \prod_{k=1}^{n}(M_k+1) = \prod_{k=1}^{n}(m+1) = (m+1)^n \tag{A.5}$$

Now, we will demonstrate that the expressions for the number of columns (13) and rows (A.8) in the coefficient list $c_{i_j,j}$ are correct for the CHSH inequality. In this case, we have $M_k = 2$ and $n = 2$. For the number of columns, When every party doesn't make a measurement $m_r = 0$, we end up with the first column $j = 1$ as the expression disappears. Now consider the case that both parties make their second measurement $m_r = 2$, we expect the last of the 9 columns.

$$j = \sum_{\substack{r=1 \\ m_r \in \mathcal{M}}}^{n}\left(m_r \prod_{k=r+1}^{n}(M_k+1)\right)+1 = \sum_{r=1}^{2}\left(2\prod_{k=r+1}^{2}(2+1)\right)+1 = 2\cdot 3 + 2\cdot 1 + 1 = 9 \tag{A.6}$$

Now consider the case that the first party makes their first measurement $m_1 = 1$ and the second party makes their second measurement $m_2 = 2$. This represents column 6.

$$j = \sum_{\substack{r=1 \\ m_r \in \mathcal{M}}}^{2}\left(m_r \prod_{k=r+1}^{2}(2+1)\right)+1 = 1\cdot 3 + 2\cdot 1 + 1 = 6 \tag{A.7}$$

Since all the parties have the same number of outcomes, the number of rows of the coefficient list for CHSH depends only on the number of parties that make measurements $n' \leq n$.

$$R_j = \prod_{\substack{r \\ \forall\{r\,|\,m_r \in \mathcal{M}\,,\, m_r \neq 0\}}} D_{m_r,r} = \prod_{k=1}^{n'} d = d^0 \text{ or } d^1 \text{ or } d^2 = 1 \text{ or } 2 \text{ or } 4 \tag{A.8}$$

References

[1] Jing-Ling Chen and Dong-Ling Deng. Bell inequality for qubits based on the cauchy-schwarz inequality. *Phys. Rev. A*, 79:012115, 2009.

[2] Jing-Ling Chen and Dong-Ling Deng. Tight correlation-function bell inequality for multipartite d-dimensional systems. *Phys. Rev. A*, 79:012111, Jan 2009.

[3] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, 1969.

[4] Daniel Collins and Nicolas Gisin. A relevant two qubit bell inequality inequivalent to the chsh inequality. *Journal of Physics A: Mathematical and General*, 37(5):1775, 2004.

[5] S. Pironio. *Aspects of quantum non-locality*. PhD thesis, Université Libre de Bruxelles, 2004.

[6] Neil C. Smith. An algorithm to calculate the facet dimension of a bell inequality. July 2017.

[7] Chunfeng Wu, Jing-Ling Chen, L. C. Kwek, and C. H. Oh. Correlation-function bell inequality with improved visibility for three qubits. *Phys. Rev. A*, 77:062309, 2008.